ON VANISHING VARIANCE IN TRANSFORMER LENGTH GENERALIZATION

Ruining Li*
University of Oxford
ruining@robots.ox.ac.uk

Gabrijel Boduljak*
University of Oxford
gabrijel@robots.ox.ac.uk

Jensen (Jinghao) Zhou
University of Oxford
jinghao@robots.ox.ac.uk

ABSTRACT

It is a widely known issue that Transformers, when trained on shorter sequences, fail to generalize robustly to longer ones at test time. This raises the question of whether Transformer models are real *reasoning* engines, despite their impressive abilities in mathematical problem solving and code synthesis. In this paper, we offer a *vanishing variance* perspective on this issue. To the best of our knowledge, we are the first to demonstrate that even for today's frontier models, a longer sequence length results in a decrease in variance in the output of the multi-head attention modules. On the argmax retrieval and dictionary lookup tasks, our experiments show that applying layer normalization after the attention outputs leads to significantly better length generalization. Our analyses attribute this improvement to a reduction—though not a complete elimination—of the distribution shift caused by vanishing variance.

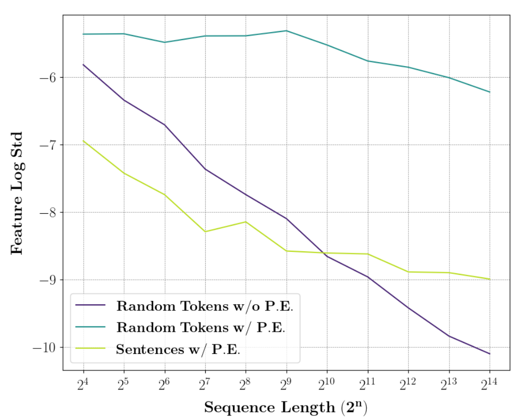

Figure 1: **Standard deviation of a fixed component in attention outputs from the first layer of Llama-3.2-1B (log-log scale) over multiple input sequences of fixed length N.** Even in the latest LLMs, increasing sequence length N reduces the variance of attended outputs, significantly degrading accuracy on long sequences.

1 BACKGROUND: VANISHING VARIANCE

It is no exaggeration to say that Transformers (Vaswani et al., 2017) is the most important architecture in modern deep learning. It is widely adopted in almost every domain, ranging from natural language (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020) and vision (Dosovitskiy et al.,

*Equal contribution; each reserves the right to be listed first.

2021; Peebles & Xie, 2023) to audio (Radford et al., 2023) and protein design (Jumper et al., 2021). Despite its successes, recent studies (Press et al., 2022; Zhou et al., 2023; 2024; Kazemnejad et al., 2024; Veličković et al., 2024) in large language models (LLMs) have shown that transformer-based models often struggle with length generalization, an ability that requires the model to generalize to longer sequences than seen during training. Several prior works have proposed to either refine position encodings (Ruoss et al., 2023; Zhou et al., 2024; Kazemnejad et al., 2024) or adapt the softmax function (Press et al., 2022; Veličković et al., 2024) to improve length generalization. However, these methods are ad-hoc and lack interpretability, making it more of an art than a science to understand when and why they work.

In this paper, we study the distribution shift that occurs in the intermediate outputs when an attention module trained on shorter sequences is subsequently exposed to longer ones in a zero-shot manner. We hope that our findings will encourage future research on network architectures that are provably robust (*e.g.*, invariant) to varying sequence lengths.

Background and notations. At the core of Transformers is the attention mechanism (Vaswani et al., 2017). The attention first projects the input sequence $\mathbf{X} = [\mathbf{x}_1 \| \mathbf{x}_2 \| \dots \| \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$, where N is the sequence length and each item $\mathbf{x}_n \in \mathbb{R}^D$ has D features, into keys $\mathbf{K} = \mathbf{X}\mathbf{W}_K \in \mathbb{R}^{N \times D}$ and values $\mathbf{V} = \mathbf{X}\mathbf{W}_V \in \mathbb{R}^{N \times D}$ using learnable weight matrices $\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D \times D}$. Similarly, the query sequence $\mathbf{Y} = [\mathbf{y}_1 \| \mathbf{y}_2 \| \dots \| \mathbf{y}_M]^\top \in \mathbb{R}^{M \times D}$ is projected into queries $\mathbf{Q} = \mathbf{Y}\mathbf{W}_Q \in \mathbb{R}^{M \times D}$ using $\mathbf{W}_Q \in \mathbb{R}^{D \times D}$. The attention then computes $\mathbf{O} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}}\right)\mathbf{V} \in \mathbb{R}^{M \times D}$ and projects it using another weight matrix $\mathbf{W}_O \in \mathbb{R}^{D \times D}$ to yield the final result $\mathrm{Attn}(\mathbf{X}, \mathbf{Y}) = \mathbf{O}\mathbf{W}_O$. In this paper, we use the term "attention weights" to refer to the softmax score, *i.e.*, $\mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}})$, "attention outputs" the intermediate $\mathbf{O}$, and $\mathrm{Attn}(\mathbf{X}, \mathbf{Y})$ the final result.

Our main observation is the *vanishing variance* problem: as the sequence length N increases, the variance of attention outputs (computed over multiple input sequences of length N) decreases. We formalize this as Proposition 1.

Proposition 1 (The vanishing variance problem)**.** *Consider a **trained** attention module with weights* $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$. *Let* $\mathbf{X} = [\mathbf{x}_1 \| \mathbf{x}_2 \| \dots \| \mathbf{x}_N]^\top$ *denote an input sequence of length* N. *If (1)* $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \overset{i.i.d}{\sim} \mathcal{X}$, *a distribution over a **finite** vocabulary, and (2)* $\mathbb{E}_{\mathbf{x} \sim \mathcal{X}}[\mathbf{W}_V \mathbf{x}] = \mathbf{0}$, *then for a **fixed** query* $\mathbf{y}$ *and a **fixed** feature* d,

$$\lim_{N \to \infty} \mathrm{Var}_{(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \sim \mathcal{X}^N} \left(\left[\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}} \right) \mathbf{V} \right]_d \right) = 0,$$

where $\mathbf{x}_n, \mathbf{y} \in \mathbb{R}^D$ *and* $\mathbf{Q} \in \mathbb{R}^{1 \times D}, \mathbf{K} \in \mathbb{R}^{N \times D}, \mathbf{V} \in \mathbb{R}^{N \times D}$ *are intermediate results in* $\mathrm{Attn}(\mathbf{X}, [\mathbf{y}])$.

Informally, for a fixed component d *in the attention outputs, its variance over input sequences of length* N*, where each sequence consists of* N *independently and identically distributed (i.i.d.) tokens, vanishes as* $N \to \infty$.

Proof. Please refer to Appendix A. □

Note that assumptions of Proposition 1 are violated in practice. In particular, the independence assumption does not hold in LLMs because of (1) the introduction of positional encoding, and more significantly (2) the nature of language, where preceding words provide important context for those that follow. In addition, $\mathbb{E}[\mathbf{W}_V \mathbf{x}_i] = \mathbf{0}$ is not strictly enforced. Nevertheless, we find that even for today's frontier LLMs, the decay in attention output variance, as established in Proposition 1, remains pronounced. In Fig. 1, we plot the standard deviation σ of a fixed component of the attention outputs from the first layer of Llama-3.2-1B (AI@Meta, 2024) as a function of input sequence length N. σ is computed over 100 length-N sequences sampled randomly with 3 strategies: (Random Tokens w/o P.E.) We sample single tokens i.i.d uniformly at random from the tokenizer's vocabulary, and remove the positional encoding for inference; (Random Tokens w/ P.E.) We still sample single tokens i.i.d uniformly at random, but keep the positional encoding at inference time; (Sentences w/ P.E.) We sample consecutive sentences from a long paragraph[1], and truncate the token sequences to length

[1] Obtained from https://github.com/dscape/spell/blob/master/test/resources/big.txt

N—such sequences lie *within* the LLM's training distribution. As can be seen in the log-log plot, for Random Tokens w/o P.E., where the independence assumption *does* hold, σ scales with input sequence length N roughly as $\sigma \propto N^{-0.5}$. For Random Tokens w/ P.E. and Sentences w/ P.E., where such assumption is no longer valid, the downward trend is still obvious.

2 Layer Normalization for Length Generalization

As variance vanishes with longer sequence lengths in attention outputs, we are intrigued to investigate the causes of performance degradation observed in LLMs. To this end, we perform a toy study on the statistical behavior of attention output values.

For simplicity, we consider a one-layer Transformer with single-head attention, omitting residual connections and normalization, following Veličković et al. (2024). We adopt this architecture as our *Baseline*. The model receives a *single* query token and an input sequence of varying length to perform simple algorithmic tasks. To eliminate confounds from positional encodings, we focus on order-invariant tasks, where the output depends only on the multiset (not the order) of input tokens, including argmax retrieval and dictionary lookup. Our goal is to evaluate models trained on shorter sequences using longer (*i.e.*, out-of-distribution in length) sequences to study length generalization. More details of the model architecture and synthetic data generation are provided in Appendix B.

In Fig. 2, we visualize the distribution of 5 individual components in attention outputs $\mathbf{O}$ across multiple input sequences of lengths 2^4, 2^{12} and 2^{14}, obtained with a model checkpoint trained on sequences of up to length 2^4. As can be seen in the top row, testing on out-of-distribution sequence lengths leads to vanishing variance, causing a distribution shift where each individual component of $\mathbf{O}$ becomes more concentrated around its mean.

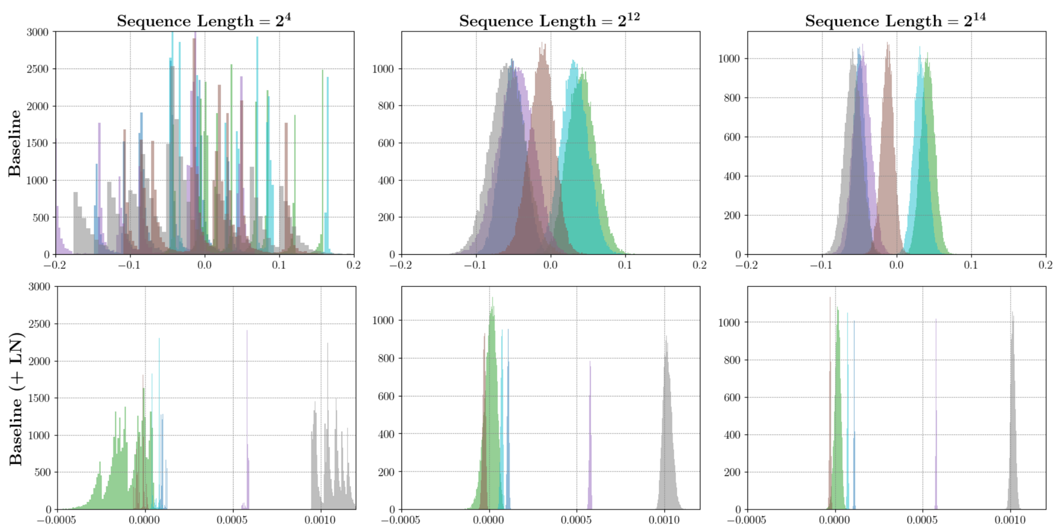

Figure 2: **Distribution of 5 individual features in attention outputs O across batches.** Each color represents a different feature. As input sequence length N increases from 2^4 to 2^{14}, feature variance decreases, and values concentrate around their mean. Layer normalization (bottom) scales and shifts features to maintain relatively constant *global* variance, likely explaining its superior length generalization compared to the *Baseline* (top).

While this distribution shift of individual features is expected according to Proposition 1, we are more interested in the distribution shift of the entire feature vector in $\mathbb{R}^D$, as the whole vector is subsequently input to an MLP to predict the final result. As input sequence length N increases, each feature is less likely to have extreme values (as its distribution is more centered). Consequently, the *global* feature variance, defined as $\sigma^2_{\text{global}} = \frac{1}{D}\Sigma_{d=1}^{D}(\mathbf{o}_d - \mu_{\text{global}})^2$ where $\mu_{\text{global}} = \frac{1}{D}\Sigma_{d=1}^{D}\mathbf{o}_d$ is the global mean, also decreases. We illustrate this observation in Fig. 3 (right), where the global variance decays as N increases. In Fig. 3 (left), we show that in addition to the global variance, the global mean μ_{global} also exhibits drift. Such a distribution shift in attention outputs (and thus MLP inputs)

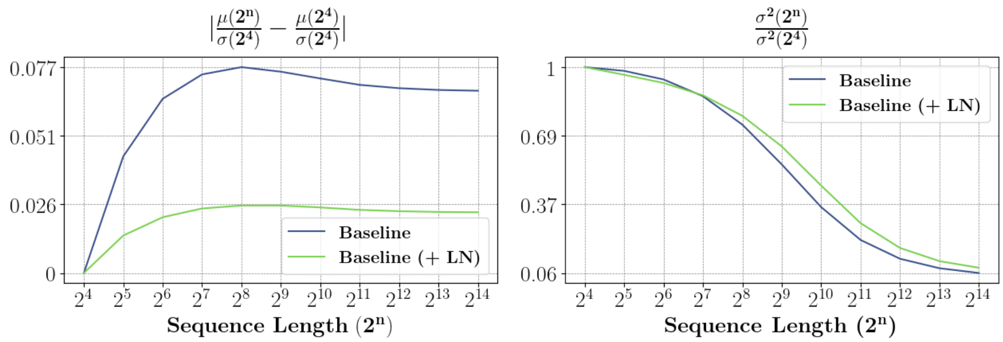

Figure 3: **Layer normalization helps mitigate distribution shift in attention outputs.** (**Left**) shows the drift in global mean as input sequence length deviates from the training distribution. The mean is normalized by the training global variance to eliminate scale differences. (**Right**) shows the decay in global variance. All results are averaged across 32k random input sequences of the fixed length.

hinders generalization, since the MLP is only trained on features with larger global variance and a different global mean (Zhou et al., 2022).

To mitigate this distribution shift, we explore applying layer normalization (Ba et al., 2016) immediately after the attention outputs, *i.e.*, $\mathrm{LayerNorm}(\mathbf{O})_{b,t,d} = \gamma_d \cdot \frac{\mathbf{o}_{b,t,d} - \mu_{b,t}}{\sigma_{b,t} + \epsilon} + \beta_d$, where $\mathbf{O}$ is the batched attention outputs, $\mu_{b,t} = \frac{1}{D}\Sigma_{d=1}^{D}\mathbf{O}_{b,t,d}$, $\sigma_{b,t} = \sqrt{\frac{1}{D}\Sigma_{d=1}^{D}(\mathbf{O}_{b,t,d} - \mu_{b,t})^2}$, and $\gamma, \beta \in \mathbb{R}^D$ are learnable scale and shift parameters. While variance decay in individual features is inevitable (bottom row of Fig. 2), standardization and learnable scale and shift parameters help stabilize the feature distribution. This adjustment preserves the global mean and variance more effectively as sequence length increases $1000\times$ (Fig. 3). This enhances length generalization, as discussed next.

3 Experiments

We consider *two* tasks: argmax retrieval and dictionary lookup. The former has been considered by Veličković et al. (2024). The latter closely resembles the core function of the attention mechanism (*i.e.*, to retrieve the most relevant information based on the similarity between queries and keys). As detailed in Appendix B, the order of input tokens does not affect the target output in either task. By deliberately selecting such tasks, we isolate and examine the length generalization capabilities of the attention mechanism itself, independent of any effects introduced by positional encodings (Zhou et al., 2024). We generate synthetic data (of input sequence length up to 16) to train the models, and evaluate them on sequences of length up to 2^{14}.

3.1 Results and Analysis

The results presented in Table 1 and Table 2 indicate that applying layer normalization to attention outputs leads to consistently better accuracy on out-of-distribution sequence lengths, with statistical significance confirmed by a paired t-test over 100 training runs from different random seeds.

Test-time adaptation and fine-tuning are common techniques for improving length generalization in transformers (Anil et al., 2022; Veličković et al., 2024). To show that the benefits of layer normalization are *orthogonal* to these techniques, we implement the adaptive temperature method from Veličković et al. (2024) in both architectures, with and without layer normalization. Combined with test-time adaptation, layer normalization still yields a significant improvement. In Fig. 4, we demonstrate that layer normalization also mitigates dispersion (Veličković et al., 2024).

Does layer normalization alleviate distribution shift? Layer normalization does alleviate—but *not* eliminate—distribution shift. With layer normalization, the global mean and global variance remain more stable on out-of-distribution sequence lengths (Fig. 3). However, the variance of fixed components in attention outputs still decays, regardless of layer normalization (Fig. 2).

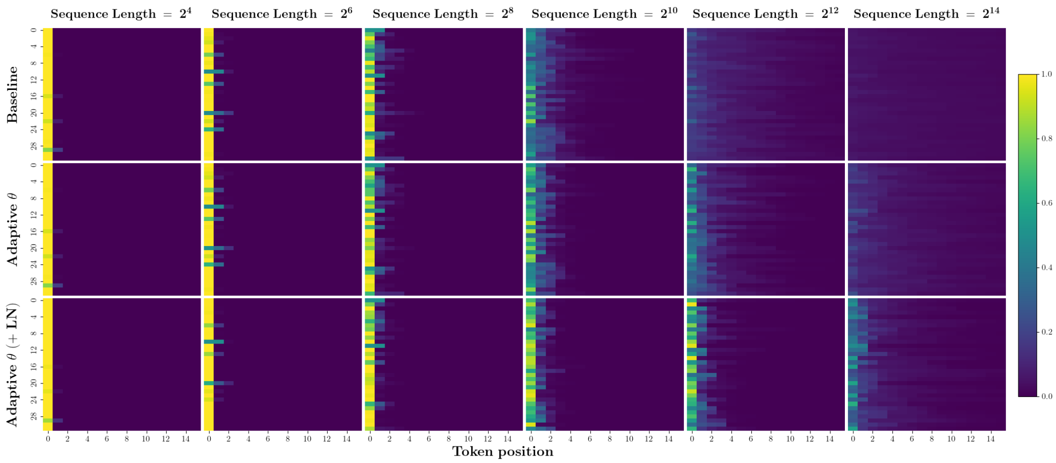

Figure 4: **Heatmap** of the largest 16 attention weights, computed over 32 examples. Layer normalization mitigates dispersion, which is inevitable as sequence length increases (Veličković et al., 2024).

Table 1: **Results (%) on the** argmax **retrieval task.** Results are averaged over 100 runs with different random seeds. p-values are computed using a paired t-test. Entries highlighted in green indicate those with in-distribution sequence lengths.

Model	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
w.o. test-time adaptation											
Baseline	**99.6**	99.2	98.4	96.8	93.7	88.0	78.0	62.5	44.2	29.7	20.8
Baseline (+ LN)	**99.6**	**99.3**	**98.6**	**97.4**	**94.8**	**89.8**	**81.0**	**66.9**	**49.2**	**33.0**	**22.6**
p-value	$8/10^1$	$4/10^2$	$4/10^2$	$2/10^2$	$4/10^5$	$2/10^4$	$1/10^4$	$2/10^5$	$2/10^6$	$4/10^5$	$3/10^3$
w. test-time adaptation											
Adaptive θ	99.6	99.2	98.5	96.9	94.1	89.1	81.2	69.1	54.2	39.0	27.1
Adaptive θ (+ LN)	**99.7**	**99.4**	**98.7**	**97.5**	**95.1**	**91.0**	**84.0**	**73.6**	**58.9**	**43.1**	**30.4**
p-value	$7/10^1$	$5/10^4$	$1/10^2$	$5/10^5$	$4/10^4$	$5/10^5$	$1/10^4$	$2/10^5$	$8/10^5$	$4/10^4$	$7/10^4$

3.2 Ablations

In addition to layer normalization, we explore an alternative normalization strategy in which we *standardize* (*i.e.*, std. in Table 3) the attention outputs across the D features without the learnable scale and shift parameters present in LN, *i.e.*, $\mathrm{Standardize}(\mathbf{O})_{b,t,d} = \frac{\mathbf{O}_{b,t,d} - \mu_{b,t}}{\sigma_{b,t} + \epsilon}$, where $\mu_{b,t}$ and $\sigma_{b,t}$ are computed in the same manner as in layer normalization.

Table 2: **Results (%) on the dictionary lookup task**. Results are averaged over 100 runs with different random seeds. p-values are computed using a paired t-test. Entries highlighted in green indicate those with in-distribution sequence lengths.

Model	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
w.o. test-time adaptation											
Baseline	99.3	98.6	97.3	94.7	89.5	80.4	67.6	52.9	38.7	26.5	17.8
Baseline (+ LN)	**99.4**	**98.8**	**97.6**	**95.3**	**90.7**	**82.9**	**71.7**	**57.7**	**44.1**	**32.3**	**22.4**
p-value	$6/10^2$	$1/10^2$	$5/10^2$	$2/10^3$	$2/10^4$	$3/10^8$	$1/10^{11}$	$2/10^{12}$	$7/10^{15}$	$3/10^{21}$	$2/10^{19}$
w. test-time adaptation											
Adaptive θ	99.3	98.6	97.2	94.5	89.3	80.4	67.8	52.6	38.6	27.3	20.8
Adaptive θ (+ LN)	**99.4**	**98.8**	**97.6**	**95.4**	**90.6**	**82.9**	**71.7**	**57.8**	**44.5**	**33.4**	**27.7**
p-value	$6/10^1$	$5/10^2$	$3/10^2$	$1/10^4$	$3/10^4$	$1/10^5$	$8/10^9$	$6/10^{12}$	$2/10^{16}$	$9/10^{20}$	$1/10^{21}$

As shown in Table 3, where the relative accuracy gain over *Baseline* on the $\arg\max$ retrieval task is reported, standardization improves length generalization, even though it strictly constrains model capacity. This underscores the importance (and potential benefits) of addressing the observed distribution shift. LN outperforms standardization, as confirmed by the paired t-test. Similar ablation results on the dictionary lookup task can be found in Appendix B.3.

Table 3: **Ablations** on different normalization strategies on the argmax retrieval task. Relative results (%) compared to the *Baseline* ($\triangle$) are reported.

Model	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
$\triangle$ (+ std.)	-0.05	$+0.00$	$+0.09$	$+0.26$	$+0.60$	$+0.84$	$+1.62$	$+2.26$	$+3.05$	$+1.80$	$+0.70$
$\triangle$ (+ LN)	$\mathbf{+0.01}$	$\mathbf{+0.11}$	$\mathbf{+0.21}$	$\mathbf{+0.57}$	$\mathbf{+1.15}$	$\mathbf{+1.81}$	$\mathbf{+2.98}$	$\mathbf{+4.32}$	$\mathbf{+4.99}$	$\mathbf{+3.30}$	$\mathbf{+1.76}$
p-value	$2/10^3$	$7/10^4$	$1/10^2$	$5/10^4$	$3/10^4$	$3/10^4$	$7/10^4$	$2/10^4$	$3/10^4$	$1/10^3$	$5/10^3$

4 Related Work

Positional encoding for length generalization. Many works have attributed the inability of Transformers to extrapolate to longer sequences to positional encoding. Several alternatives to the sinusoidal positional encoding originally introduced by Vaswani et al. (2017) have been proposed to enhance the performance of Transformer-based models in natural language processing (NLP) tasks, including relative positional encoding (Shaw et al., 2018; Dai et al., 2019), rotary positional encoding (Su et al., 2024), no positional encoding (Haviv et al., 2022) and randomized positional encoding (Ruoss et al., 2023). Authors have examined the impact of different variants of positional encoding on length generalization (Chi et al., 2022; Ruoss et al., 2023; Kazemnejad et al., 2024; Li et al., 2024; Peng et al., 2024; Zhou et al., 2024). Unlike prior work that explores positional encoding for length generalization, we focus on algorithmic tasks that are order-invariant. We present a *vanishing variance* perspective on length generalization which is orthogonal to the extrapolability of positional encoding.

Alternatives to softmax attention. The softmax output has been utilized to interpret the inner workings of Transformers (Xu et al., 2015; Choi et al., 2016; Martins & Astudillo, 2016). More recently, Veličković et al. (2024) demonstrated that the attention weights output by softmax will *disperse* as sequence length increases, attributing this phenomenon to the Transformer's limited capability in length generalization. In this paper, we show that this dispersion leads to the vanishing variance problem in the intermediate attention outputs. While many variants of softmax attention have been introduced (Correia et al., 2019; Press et al., 2022; Tan et al., 2024; Ye et al., 2024), they are motivated mostly by interpretability, rather than the distribution of attention outputs for length generalization. To the best of our knowledge, none of the existing works have fundamentally eliminated the vanishing variance problem we presented in this paper. We hope our study can motivate designs of network architectures that are provably invariant to sequence length variations.

5 Conclusion

In this paper, we have introduced the *vanishing variance* problem and provided both theoretical analysis and empirical evidence demonstrating its role in inducing distribution shift in attention outputs. This shift hinders the ability of Transformers to generalize effectively to out-of-distribution sequence lengths. We demonstrated that mitigating this distribution shift through techniques like layer normalization and standardization—despite potential trade-offs in model expressiveness—significantly improves length generalization in attention models.

Future work. We conduct our experiments using a single-layer, single-head attention architecture for simplicity, while real-world models typically use multi-layer, multi-head attention. Our conclusions may not fully generalize to these more complex architectures. Future work may validate the normalization strategies on larger benchmarks like CLRS (Veličković et al., 2022) and real-world LLMs. Moreover, layer normalization only *partially* mitigates distribution shift presented in this paper, and is already widely adopted in Transformers (though not immediately after attention outputs). Future work may design architectures that are provably invariant to sequence length variations.

REFERENCES

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md. 2, 11

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556, 2022. 4

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 1

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. *NeurIPS*, 2022. 6

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *NeurIPS*, 2016. 6

Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. *arXiv preprint arXiv:1909.00015*, 2019. 6

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, 2019. 6

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information. In *EMNLP*, 2022. 6

John M. Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andy Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 2021. 2

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *NeurIPS*, 2024. 2, 6

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontanon, Manzil Zaheer, Sumit Sanghai, Yiming Yang, Sanjiv Kumar, and Srinadh Bhojanapalli. Functional interpolation for relative positions improves long context transformers. In *ICLR*, 2024. 6

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *ICML*, 2016. 6

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *ICLR*, 2024. 6

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*, 2022. 2, 6

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *ICML*, 2023. 2

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. Randomized positional encodings boost length generalization of transformers. In *ACL*, 2023. 2, 6

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 6

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 6

Shawn Tan, Yikang Shen, Songlin Yang, Aaron Courville, and Rameswar Panda. Stick-breaking attention. *arXiv preprint arXiv:2410.17980*, 2024. 6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 6

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The clrs algorithmic reasoning benchmark. In *ICML*, 2022. 6

Petar Veličković, Christos Perivolaropoulos, Federico Barbero, and Razvan Pascanu. softmax is not enough (for sharp out-of-distribution). *arXiv preprint arXiv:2410.01104*, 2024. 2, 3, 4, 5, 6, 10

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 6

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024. 6

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*, 2023. 2

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *PAMI*, 2022. 4

Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. Transformers can achieve length generalization but not robustly. *arXiv preprint arXiv:2402.09371*, 2024. 2, 4, 6

A Proof of Proposition 1

Proposition 1 (The vanishing variance problem). *Consider a **trained** attention module with weights* $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O$. *Let* $\mathbf{X} = [\mathbf{x}_1 \| \mathbf{x}_2 \| \dots \| \mathbf{x}_N]^\top$ *denote an input sequence of length* N. *If (1)* $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \overset{i.i.d}{\sim} \mathcal{X}$, *a distribution over a **finite** vocabulary, and (2)* $\mathbb{E}_{\mathbf{x}\sim\mathcal{X}}[\mathbf{W}_V \mathbf{x}] = \mathbf{0}$, *then for a **fixed** query* $\mathbf{y}$ *and a **fixed** feature* d,

$$\lim_{N\to\infty} \mathrm{Var}_{(\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_N)\sim\mathcal{X}^N} \left(\left[\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}} \right) \mathbf{V} \right]_d \right) = 0,$$

where $\mathbf{x}_n, \mathbf{y} \in \mathbb{R}^D$ *and* $\mathbf{Q} \in \mathbb{R}^{1\times D}, \mathbf{K} \in \mathbb{R}^{N\times D}, \mathbf{V} \in \mathbb{R}^{N\times D}$ *are intermediate results in* $\mathrm{Attn}(\mathbf{X}, [\mathbf{y}])$.

Informally, for a fixed component d *in the attention outputs, its variance over input sequences of length* N*, where each sequence consists of* N *independently and identically distributed (i.i.d.) tokens, vanishes as* $N \to \infty$.

Proof. Let $\mathbf{V}_{n,d} = [\mathbf{W}_V \mathbf{x}_n]_d$. Firstly, we argue that $\mathbf{V}_{k,d}$ and $\mathbf{V}_{l,d}$ are independent for $k \neq l$. Let $\pi_d : \mathbb{R}^D \to \mathbb{R}$ be the projection onto the d-th coordinate, namely $\pi_d(\mathbf{v}) = \mathbf{v}_d$ for $\mathbf{v} \in \mathbb{R}^D$. Observe that $\mathbf{V}_{k,d} = \pi_d(\mathbf{W}_V \mathbf{x}_k) = (\pi_d \circ \mathbf{W}_V)(\mathbf{x}_k)$ and $\mathbf{V}_{l,d} = \pi_d(\mathbf{W}_V \mathbf{x}_l) = (\pi_d \circ \mathbf{W}_V)(\mathbf{x}_l)$. By assumption (1), $\mathbf{x}_k$ and $\mathbf{x}_l$ are independent for $k \neq l$. Since $\mathbf{V}_{k,d}$, $\mathbf{V}_{l,d}$ are measurable functions of independent random variables, they are independent for $k \neq l$. By assumption, $\mathbf{x}_k$ and $\mathbf{x}_l$ are identically distributed for every k, l, so $\mathbf{V}_{k,d}$ and $\mathbf{V}_{l,d}$ are identically distributed. Thus, $\mathrm{Var}[\mathbf{V}_{k,d}]$ depends only on d, not on k. Set $\sigma_d^2 = \mathrm{Var}[\mathbf{V}_{k,d}]$. σ_d^2 is finite since the vocabulary is finite and thus compact and bounded.

Let $\mathbf{A}$ denote the attention weights $\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}}\right) \in \mathbb{R}^{1\times N}$ as the query sequence $[\mathbf{y}]$ consists of only a *single* item. Let $\mathbf{A}_n$ denote the n-th element of $\mathbf{A}$. We have

$$\begin{aligned}
\mathrm{Var}\left(\sum_{n=1}^N \mathbf{A}_n \mathbf{V}_{n,d}\right) &= \mathbb{E}\left[\left(\sum_{n=1}^N \mathbf{A}_n \mathbf{V}_{n,d}\right)^2\right] - \left(\mathbb{E}\left[\sum_{n=1}^N \mathbf{A}_n \mathbf{V}_{n,d}\right]\right)^2 \\
&\leq \mathbb{E}\left[\left(\sum_{n=1}^N \mathbf{A}_n \mathbf{V}_{n,d}\right)^2\right] \\
&= \mathbb{E}\left[\sum_{n=1}^N \mathbf{A}_n^2 \mathbf{V}_{n,d}^2\right] + \mathbb{E}\left[\sum_{1\leq k,l\leq N, k\neq l} \mathbf{A}_k \mathbf{A}_l \mathbf{V}_{k,d} \mathbf{V}_{l,d}\right] \\
&\leq \mathbb{E}\left[\sum_{n=1}^N (\max_{1\leq n\leq N} \mathbf{A}_n)^2 \mathbf{V}_{n,d}^2\right] + \mathbb{E}\left[\sum_{1\leq k,l\leq N, k\neq l, \mathbf{V}_{k,d}\mathbf{V}_{l,d}\geq 0} \mathbf{A}_k \mathbf{A}_l \mathbf{V}_{k,d} \mathbf{V}_{l,d}\right] \\
&\leq (\max_{1\leq n\leq N} \mathbf{A}_n)^2 \sum_{n=1}^N \mathbb{E}\left[\mathbf{V}_{n,d}^2\right] + \mathbb{E}\left[\sum_{1\leq k,l\leq N, k\neq l, \mathbf{V}_{k,d}\mathbf{V}_{l,d}\geq 0} \mathbf{V}_{k,d} \mathbf{V}_{l,d}\right] \\
&\leq N(\max_{1\leq n\leq N} \mathbf{A}_n)^2 \sigma_d^2 + \sum_{1\leq k,l\leq N, k\neq l, \mathbf{V}_{k,d}\mathbf{V}_{l,d}\geq 0} \mathbb{E}\left[\mathbf{V}_{k,d} \mathbf{V}_{l,d}\right] \\
&\leq N(\max_{1\leq n\leq N} \mathbf{A}_n)^2 \sigma_d^2 + \sum_{1\leq k,l\leq N, k\neq l, \mathbf{V}_{k,d}, \mathbf{V}_{l,d}\geq 0} \mathbb{E}\left[\mathbf{V}_{k,d}\right] \mathbb{E}\left[\mathbf{V}_{l,d}\right] \\
&\leq N(\max_{1\leq n\leq N} \mathbf{A}_n)^2 \sigma_d^2
\end{aligned}$$

In the derivation above, we used the fact that $\mathbf{A}_n \in [0, 1]$ and that for $k \neq l$, $\mathbf{V}_{k,d}$ and $\mathbf{V}_{l,d}$ are independent. We also used the assumption that for every $1 \leq k \leq N$, $\mathbb{E}\left[\mathbf{V}_{k,d}\right] = 0$. Since the tokens come from a finite dictionary, and since $\mathbf{x} \to \mathbf{W}_Q \mathbf{x}$ and $\mathbf{x} \to \mathbf{W}_K \mathbf{x}$ are continuous functions on compact domain (dictionary is finite), the logits $\langle \mathbf{W}_Q \mathbf{x}_i, \mathbf{W}_K \mathbf{x}_j \rangle$ are bounded, because they are

Table 4: **Results (%) on the** $\arg\max$ **retrieval task.** Results are averaged over 100 runs with different random seeds. p-values are computed using a paired t-test. Entries highlighted in green indicate those with in-distribution sequence lengths.

Model	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
Baseline	99.8	99.8	99.6	99.3	98.5	97.1	94.4	89.1	79.9	65.8	47.8
Baseline (+ LN)	**99.9**	**99.9**	**99.8**	**99.5**	**99.1**	**98.1**	**96.2**	**92.8**	**86.3**	**75.1**	**58.7**
p-value	$1/10^8$	$6/10^6$	$1/10^4$	$2/10^4$	$9/10^8$	$5/10^8$	$3/10^8$	$5/10^{10}$	$1/10^{11}$	$3/10^{12}$	$1/10^{13}$

Table 5: **Results (%) on the dictionary lookup task**. Results are averaged over 100 runs with different random seeds. p-values are computed using a paired t-test. Entries highlighted in green indicate those with in-distribution sequence lengths.

Model	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
Baseline	**99.9**	**99.9**	99.8	99.7	99.5	99.1	98.3	96.5	93.5	87.7	77.8
Baseline (+ LN)	**99.9**	**99.9**	**99.9**	**99.8**	**99.6**	**99.3**	**98.7**	**97.5**	**95.0**	**90.4**	**82.6**
p-value	$2/10^1$	$1/10^2$	$5/10^2$	$2/10^3$	$3/10^3$	$6/10^5$	$4/10^7$	$8/10^9$	$2/10^{10}$	$2/10^{10}$	$2/10^{13}$

continuous image of a compact set and every compact set on the real line is closed and bounded. By Lemma 2.1 of Veličković et al. (2024), there exist a constant $C > 0$ and $N_0 \in \mathbb{N}$, such that for every $N \geq N_0$, $(\max_{1\leq n\leq N} \mathbf{A}_n)^2 < \frac{C}{N^2}$. Then for every $N \geq N_0$,

$$\mathrm{Var}\left(\sum_{n=1}^{N} \mathbf{A}_n \mathbf{V}_{n,d}\right) \leq N\sigma^2 \frac{C}{N^2} = \sigma^2 \frac{C}{N}.$$

Let $\epsilon > 0$. There exists $N_1 \in \mathbb{N}$ such that for every $N \geq N_1$, $\sigma^2 \frac{C}{N} < \epsilon$. Then for every $N \geq \max(N_0, N_1)$,

$$\mathrm{Var}\left(\sum_{n=1}^{N} \mathbf{A}_n \mathbf{V}_{n,d}\right) < \epsilon.$$

□

B More Experimental Details

B.1 Implementation Details

$\arg\max$ **retrieval.** We follow Veličković et al. (2024) and train the same neural network architecture in PyTorch for $100,000$ gradient steps with the same hyper-parameter setup. We also follow Veličković et al. (2024) to generate data of varying number of items to train and test the model.

dictionary **lookup.** The network architecture is the same as the $\arg\max$ retrieval task. We generate data for training and evaluation in the following way: for each item of the length-N sequence,

Table 6: **Ablations** on different normalization strategies on the dictionary lookup task. Relative results (%) compared to the *Baseline* (△) are reported.

Model	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
△ (+ std.)	**+0.09**	+0.14	+0.23	+0.54	+1.08	+2.27	+3.49	+4.78	+5.14	+5.65	**+4.57**
△ (+ LN)	**+0.09**	**+0.20**	**+0.30**	**+0.64**	**+1.22**	**+2.55**	**+4.06**	**+4.86**	**+5.38**	**+5.72**	+4.51
p-value	1.0	0.3	0.6	0.6	0.6	0.5	0.3	0.9	0.6	0.9	0.9

we sample a value class $c_V \sim \mathcal{U}\{1, \ldots, C_V\}$ i.i.d at random; each item also has a key class $1 \leq c_K \leq C_K$. The key classes of all N items in the sequence are sampled *without* replacement. In our experiments, $C_K = 16384$ and $C_V = 64$.

The features of each item $\mathbf{x}_i$ is defined as $\mathrm{Emb}_K(c_K) \parallel \mathrm{Emb}_V(c_V)$, *i.e.*, the concatenation of the embeddings of the key class and the value class. The embedding vectors for each (key and value) class are optimized jointly with the attention network.

The query sequence in our case is guaranteed to be of length 1. We sample a key class present in the input sequence and use its embedding vector as the query.

For this task, we found that the optimization usually converges within $10,000$ gradient steps. We train the attention network, together with the embedding vectors, in PyTorch for $10,000$ steps with the same hyper-parameter setup as the $\arg\max$ task.

B.2 Results When Training on More Diverse Sequence Lengths

To validate the utility of normalization when the length gap between the training sequences and the test sequences is smaller, we follow the same experimental setup as in Section 3, but sample sequences of up to 256 items during training. We found it beneficial to gradually increase the length of the sequences sampled throughout training, as is commonly done during pre-training of frontier LLMs (AI@Meta, 2024). The results are reported in Table 4 and Table 5. With layer normalization, the accuracies on out-of-distribution sequence lengths are significantly higher than without on both tasks, demonstrating the importance of normalization for length generalization over various training settings.

B.3 Ablations on the Dictionary Lookup Task

Ablation results on the dictionary lookup task are shown in Table 6, which are consistent with the results on the $\arg\max$ retrieval task presented in Section 3.2. However, on this task, the performance of standardization and layer normalization is more similar, as indicated by the larger p-values, suggesting weaker statistical evidence for a significant difference.